A class of problems of NP to be worth to search an efficient solving algorithm

Anatoly D. Plotnikov*

Abstract

We examine possibility to design an efficient solving algorithm for problems of the class NP. It is introduced a classification of NP problems by the property that a partial solution of size k can be extended into a partial solution of size $k+1$ in polynomial time. It is defined an unique class problems to be worth to search an efficient solving algorithm. The problems, which are outside of this class, are inherently exponential. We show that the Hamiltonian cycle problem is inherently exponential.

1 Introduction

A problem Z belongs to the class NP if it has a finite input of size n, a finite output (a solution) of size $p(n)$, where $p(n)$ is some polynomial, and verifying time of the solution is also a polynomial on n.

Every problem of NP is solvable in classical sense since it can be solved by a deterministic Turing machine [4].

A solving algorithm of a problem $Z \in$ NP is called *efficient* if the solution of Z can be obtained in the polynomial number of steps on n. A set of problems of NP, having a polynomial solving algorithm, is denoted by P.

One of the main achievements in designing of solving algorithms is joined with Matroid Theory.

Let R be some finite set, and Q be a non-empty set of subsets of R. A two (R, Q), satisfying property

$$\text{if } \pi_1 \in Q \text{ and } \pi_2 \subset \pi_1 \text{ then } \pi_2 \in Q \tag{1}$$

*Vinnitsa Institute of Regional Economics and Management, e-mail: aplot@tom.vinnica.ua

is called a *hereditary* system.

A hereditary system $M = (R, Q)$ is called a *matroid* when it satisfies the following property [10]:

$$\begin{array}{l} \text{if } \pi_1, \pi_2 \in Q, \text{ and } Card(\pi_2) = Card(\pi_1) + 1 \text{ then there exists} \\ \text{an element } r \in \pi_2 \setminus \pi_1 \text{ such that } \pi_1 \cup \{r\} \in Q, \end{array} \tag{2}$$

where $r \in R$.

If an optimization problem of NP satisfies properties (1) and (2) then it can be solved by means of a *greedy* algorithm. A greedy algorithm make locally optimal choices at each step in the hope that these choices will produce a globally optimal solution [2, 8, 9].

The following assertion takes place [8].

Theorem 1 *Let (R, Q) be a hereditary system. Then a greedy algorithm produces an maximum element of Q if and only if (R, Q) be a matroid.*

Unfortunately, there are problems of NP for which an efficient solving algorithm is unknown. An analysis of NP problems shows that the main difficulty is exhaustive enumeration of the elements of the solution.

The following questions raise: What are causes for appearing of exhaustive enumeration? When can we design an efficient solving algorithm?

In this paper we show that one need to analyze a mathematical model of a problem. We define a class problem of NP be worth to search an efficient solving algorithm. We show also that other problems of NP are inherently exponential. Consequently, for similar problems the finding of the efficient solving algorithm is senseless.

2 Mathematical model

Let (R, Q) be a hereditary system. Further let P be a predicate system. For each subset R_j of R this system allows to find a value of a predicate "$R_j \in Q$?" Let it is required to find a subset $R_j \subset R$ such that $R_j \in Q$.

Many important problems of NP have the similar statement.

Satisfiability Problem (SAT). Let ϕ be a Boolean expression over n variables $x_1, \ldots, x_n$ in conjunctive normal form. It is required to find values of the variables which make ϕ equal "true", that is, co-called truth assignment of variables. Let R be a set of literals r, where r is either x_i or $\bar{x}_i$ $(i = \overline{1, n})$. The literals x and $\bar{x}$ we shall call *contrary.*

Further, let $\mathcal{B}$ be a set of subsets π^* of R determining truth assignments of variables. Denote by a set Q of subsets π of R such that $\pi \in Q$ if and only if $\pi \subseteq \pi^*$, $\pi^* \in \mathcal{B}$. Obviously that a two (R, Q) is a hereditary system.

Hamiltonian Cycle Problem (HCP). Let $G = (V, E)$ be a n-vertex undirected graph. It is required to find a cycle of edges in G which includes each of the n vertices exactly once. Let $\mathcal{B}$ be a set of Hamiltonian cycles π^* of G. Denote by a set Q of subsets π of E such that $\pi \in Q$ if and only if $\pi \subseteq \pi^*$, $\pi^* \in \mathcal{B}$. It is clear that a two (E, Q) is a hereditary system.

Consider a hereditary system (R, Q).

Let $w(r_i)$ $(i = \overline{1,n})$ be an integer, is called by a *weight* of the element r_i of R. For every $\pi \in Q$ we define a sum

$$w(\pi) = \sum_{\forall r \in \pi} w(r).$$

This sum we shall call a *weight* of π.

Let it is required to find an element π^* of Q, having the maximum weight.

We have an optimization problem.

Maximum Independent Set Problems (MISP). Let $G = (V, E)$ be a n-vertex undirected graph. It is required to find a subset $\pi^* \subseteq V$, having the maximum number vertices, such that every two vertices in π^* are non-adjacent in G. A set $\pi \subseteq V$ is called independent if every two vertices in π are non-adjacent. Let Q be a set of all independent sets of vertices in of G. It is easy to see that a two (V, Q) is a hereditary system. In this problem $w(\pi) = Card(\pi)$ for any $\pi \in Q$.

Let there be a problem $Z \in$ NP which is a hereditary system (R, Q). Any element $\pi \in Q$ we shall call an *admissible* solution of Z. An inclusion maximal admissible solution π^* of Z is called *support*. An admissible solution π of Z is called *partial* if there exists a support solution π^* such that $\pi \subseteq \pi^*$. If π be some support solution of the problem Z, and π_1 be some partial solution of this problem such that $\pi_1 \subset \pi$ then the partial solution π_1 will be *own* for support solution π.

We considered a *computational* model of the problems in NP. In Complexity Theory, every problem of NP is considered as a *decision* problem. A decision problem is a computational problem whose solution is *yes* or *no* [5]. The solution of a computational problem (in given case it is some of support solutions) we may consider as "proof" that the corresponding decision problem of NP has an answer "yes". Therefore, the conception of admissible solution is more wide than the conception of "proof" for a decision problem.

3 Sequential method

Let there be a problem $Z \in$ NP. We assume that Z is a hereditary system (R, Q). The questions raises: *How* can we construct an admissible solution of Z?

A Turing machine is a generally accepted model of computation (see, for example, [1, 5]). Therefore, we may think that we have an 1-type Turing machine M_1. The machine M_1 produces symbols into cells of tape *sequentially*, that is, symbol-by-symbol. If we shall be believe that Turing machine solve the problem Z then we may consider a result of such work of M_1 at each step as an admissible solution of Z. Naturally to consider the record of a symbol on tape as construction of the next element of admissible solution.

Thus, procedure of construction of admissible solution $\pi \in Q$ is extended on the time, i.e. elements of one are obtained element-by-element.

A method of the construct of the required solution, when we are obtaining its elements by step-by-step, element-by-element, will be called *sequential*.

Let π_1, π be respectively the partial and support solutions of a problem $Z \in$ NP such that $\pi_1 \subset \pi$. Denote the construction time for these partial and support solutions of Z by $t(\pi_1)$ and $t(\pi)$ respectively. Then the following assertion takes place.

Theorem 2 *$t(\pi_1) < t(\pi)$.*

Proof. By the definition of the sequential method, every of support solutions can be obtained *after* constructing of an own partial solution. That proves Theorem 2.$\circ$

Theorem 3 *The solution of any problem $Z \in NP$ can be obtained by a sequential method.*

Proof. We believe that each problem of the class NP is solvable (see Section 1), that is, each of such problem can be solved by the deterministic Turing machine. Since this machine works sequentially, it produces the solution of a problem by step-by-step, element-by-element. Therefore Theorem 3 is true.$\circ$

Obviously, one can believe that a sequential method is a sole *general* method of the solving for every problem $Z \in$ NP.

In fact, for example, let there is necessary to find some independent set of the graph vertices. Obviously, in common case the *simultaneous choice* of a number of such vertices is impossible if the graph structure was unknown beforehand. Clearly, each subsequent vertices can be chosen only if it is known which vertices was chosen in the formed independent set before.

4 Problems without lookahead

Let π_1 be a partial solution of the problem $Z \in$ NP. By designing of the next partial solution, the problems of NP can be partitioned into two classes [7]:

- the problems for which the next partial solution $\pi_2 = \pi_1 \cup \{r\}$ can be found in polynomial time on early found partial solution by picking one element of the set $R \setminus \pi_1$;
- all other problems of NP.

That is, the problems of NP can be classified depending on the computing time of the predicate "$\pi_1 \cup \{r\} \in Q$?" for every partial solution $\pi_1 \in Q$ and for any element $r \in R \setminus \pi_1$. If such predicate can be computed in polynomial time on the problem dimension then such problem will be called *the problem without lookahead*. Otherwise the problem is called *inherently exponential*.

A set of all problems without lookahead we will denote by UF, where UF $\subseteq$ NP.

Theorem 4 *A support solution of a problem $Z \in NP$ can be found in polynomial time if and only if $Z \in UF$.*

Proof. Let there be the problem $Z \in$ NP such that $Z \in$ UF. By definition of problems without lookahead, the next partial solution of Z can be found in polynomial time. Since $\oslash \in Q$ for any $Z \in$ UF, and a support solution contains at the most n elements, where n is a problem size, then it implies polynomial construction time of the support solution.

On the other hand, let there be the problem $Z \in$ NP such which is resolved in polynomial time. Suppose that $Z \notin$ UF. In this case there exists at least one of the partial solutions of the problem Z, determined in exponential time. By the condition of Theorem 4, the support solution of Z is found in polynomial time. We have given the contradiction of the Theorems 2 and 3.$\circ$

Thus, the class UF is induced by problems of NP for which a support solution may be construct in polynomial time. Notice that this support solution may not be the global solution of the problem.

Prove that the HCP is outside of the class UF.

Consider the following optimization problem.

Let $G = (X, E)$ be an undirected graph without loops and multiple edges, where X is the vertex set of G, and E is the edge set. Simple cycles C_i, C_p $(i \neq p)$ of G are called *disjoint* if they have no common vertices. A collection $\pi = \{C_1, \ldots, C_k$ is called a *partition of G into disjoint edges and/or cycles* if

- for each pair of cycles $C_i, C_p \in \pi$ $(i \neq p)$ $X(C_i) \cap X(C_p) = \oslash$;
- and

$$\bigcup_{\forall C_i \in \pi} X(C_i) = X,$$

where $X(C_i)$ is a set of vertices belonging to the cycle (or the edge) C_i.

It is required to find a partition π^* having the minimum number of cycles (edges).

This problem is NP-complete, and it is formulated as the Minimum Vertex Disjoint Cycle Cover Problem (MVDCCP).

It is known that an admissible solution of MVDCCP can be obtained as a solution of the assignment problem (see, for example, [3, 6]) in polynomial time. Hence, MVDCCP belongs to UF. Its support solution – some partition of G into disjoint edges and/or cycles – may be constructed in polynomial time.

On the other hand, it is evident MVDCCP is not a matroid. Therefore, locally optimal choice does not guarantee that the obtained solution of the problem will be optimal.

On Fig. 1 (a), (b) the two graph partitions into disjoint cycles/edges are represented that correspond to two distinct solutions of the same assignment problem shown on Fig. 1 (c), (d) respectively. The solutions of the assignment problem had been obtained as perfect matching in a bipartite graph [3].

Thus, the following assertion is true.

Theorem 5 *MVDCCP can not be solved by a greedy algorithm.*

Lemma 1 *If a graph G be Hamiltonian then the solution of MVDCCP is a Hamiltonian cycle.*

Proof. It is evident.$\circ$

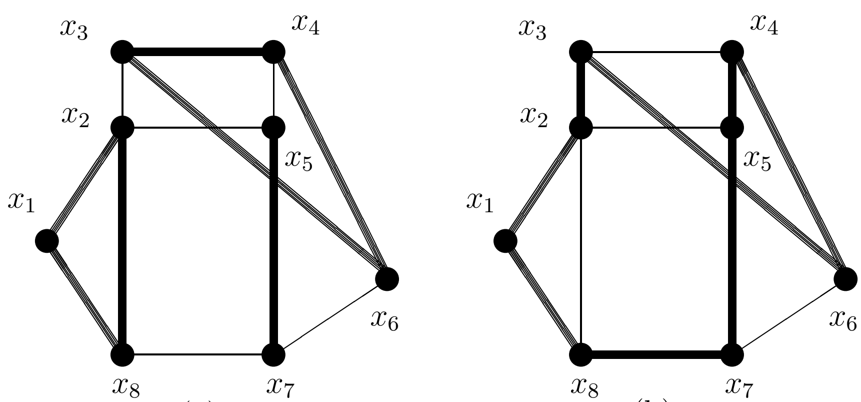
Denote edges of the graph above:

$$\begin{array}{llll}
e_1 = \{x_1, x_2\}, & e_2 = \{x_1, x_8\}, & e_3 = \{x_2, x_3\}, & e_4 = \{x_2, x_3\}, \\
e_5 = \{x_2, x_3\}, & e_6 = \{x_2, x_3\}, & e_7 = \{x_2, x_3\}, & e_8 = \{x_2, x_3\}, \\
e_9 = \{x_2, x_3\}, & e_{10} = \{x_2, x_3\}, & e_{11} = \{x_2, x_3\}, & e_{12} = \{x_2, x_3\}.
\end{array}$$

In this case, the HCP is a hereditary system (E, Q), where the set Q contains only support solution

$$\pi^* = \{e_1, e_2, e_3, e_7, e_8, e_9, e_{10}, e_{12}\}.$$

and all subsets of π^*. The set Q have no other elements.



(a) (b)

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1		1						1
x_2	1		1		1			1
x_3		1		1		1		
x_4			1		1	1		
x_5		1		1			1	
x_6			1	1			1	
x_7					1	1		1
x_8	1	1					1	

(c)

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1		1						1
x_2	1		1		1			1
x_3		1		1		1		
x_4			1		1	1		
x_5		1		1			1	
x_6			1	1			1	
x_7					1	1		1
x_8	1	1					1	

(d)

Figure 1: Two graph partitions into disjoint cycles/edges

Theorem 6 *A partial solution of HCP can not be found by a sequential method in polynomial time.*

Proof. At each step of a sequential method, we find a partial solution of a problem. Since each support solution of HCP is a Hamiltonian cycle then at each step of the sequential method we should pick an edge of some Hamiltonian cycle.

Let π_1 be some partial solution of HCP. Suppose that the next partial solution of HCP π_2, where $Card(\pi_2) = Card(\pi_1) + 1$, can be found in polynomial time. It follows that making locally optimal choice at each step – an edge of a graph – we will produce a globally optimal solution of HCP – a Hamiltonian cycle, that is, we can construct a Hamiltonian cycle by a greedy algorithm. Hence, MVDCCP can be solved by a greedy algorithm. It contradicts to Theorem 5.∘

Corollary 1 *HCP does not belong to UF.*

Thus, it is proved that the Hamilton cycle problem is inherently exponential.

Theorem 7 *If a problem of the class NP is a hereditary system then this problem is effectively solvable if and only if it belongs to the class UF.*

Proof. By Theorem 3, the solution of every problem $Z \in$ NP can be obtained by a sequential method. If the problem Z is a hereditary system then its global solution is a support solution (see Section 2). Then the validity of Theorem 7 follows from the Theorem 4.∘

Theorem 8 *Every problem of NP is reduced to some problem of UF.*

Proof. It will suffice to indicate that MISP belongs to UF.∘

Of course, if the given decision problem of NP is reduced to another decision problem of NP then we have a *new* problem. The new problem may has *other* properties than the initial problem.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms* (Addison-Wesley Publishing Company, Reading, Massachusetts, 1976).

[2] S. Baase, *Computer algorithms: Introduction to Design and Analysis.* (Addison-Wesley Publishing Company, Reading, Massachusetts, 1991).

[3] N. Christofides, *Graph theory (An Algorithmic Approach)* (Academic press, New York, 1975).

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability.* (W.H.Freeman and Company, San Francisco, 1979).

[5] Michael C. Loui, "Complexity theory," in *The Computer Science and Engineering*, Handbook, chapter 11, ed. JR Allen B. Tucker, (CRC Press, 1997) pp. 250–276.

[6] L. Lovasz, "On covering of graphs," in *Theory of Graphs*, eds. P. Erdös and G. Katona (Akad. Kiadó, 1968) pp. 231–236.

[7] A. D. Plotnikov, "Formalization of the class of problems solvable by a nondeterministic turing machine," *Cybernetics and Systems Analysis*, **33**, 5 (1997) 635–640.

[8] M. N. S. Swamy and K. Thulasiraman, *Graphs, Networks and Algorithms.* (John Wiley & Sons, N. Y., Chichester, 1981).

[9] D. B. West, *Introduction to Graph Theory.* (Prentice Hall, Inc., Upper Saddle River, NJ, 1996).

[10] R. J. Wilson, *Introduction to Graph Theory.* (Oliver and Boyd, Edinburgh, 1972).